

BAB II

LANDASAN TEORI

2.1 Teori Graf

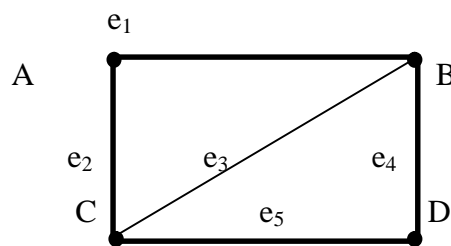
Teori graf adalah cabang kajian yang mempelajari sifat-sifat graf. Berikut adalah definisi-definisi graf dari referensi yang berbeda :

Definisi 2.1 (Munir, 2005) Graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$ yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) dan E adalah himpunan sisi (*edge* atau *arcs*) yang menghubungkan sepasang simpul.

Definisi 2.2 (Lipschut, 2002) Sebuah graf G terdiri dari dua bagian :

- i. Sebuah himpunan $V = V(G)$ memiliki elemen-elemen yang dinamakan simpul, simpul atau simpul.
- ii. Sebuah kumpulan $E = E(G)$ merupakan pasangan terurut dari simpul-simpul yang berbeda dinamakan sisi.

Kita menuliskan $G(V, E)$ bila kita ingin menyatakan dua bagian dari G . Berikut contoh Graf G :



Gambar 2.1 Graf G

Gambar 2.1 memperlihatkan graf G dengan simpul V dan sisi dengan E sebagai berikut :

$$V = \{ A, B, C, D \}$$

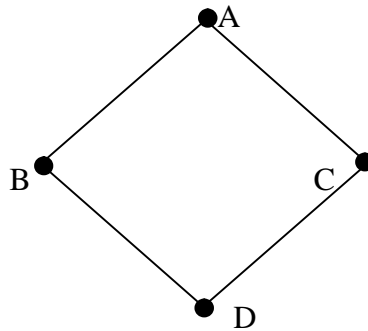
$$E = \{ (A, B), (A, C), (B, C), (B, D), (C, D) \} = \{ e_1, e_2, e_3, e_4, e_5 \}$$

2.2 Jenis-Jenis Graf

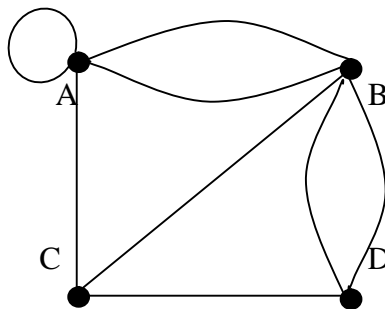
Graf dapat dikelompokkan menjadi beberapa kategori :

1. Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi ganda. Pada graf sederhana ini, sisi adalah pasangan tak berurut. Jadi untuk menuliskan sisi (u,v) sama saja dengan (v,u) .
2. Graf tidak sederhana adalah graf yang mengandung sisi ganda atau gelang.

Berikut contoh Gambar 2.2 graf sederhana dan Gambar 2.3 graf tidak sederhana :



Gambar 2.2 Graf Sederhana



Gambar 2.3 Graf Tidak Sederhana

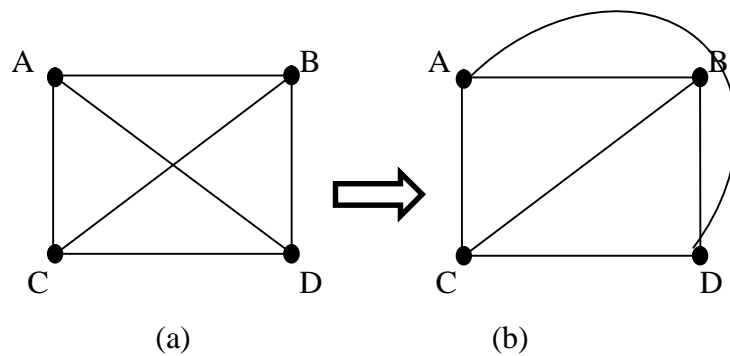
Sisi pada graf dapat mempunyai orientasi arah. Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas dua jenis :

1. Graf tak berarah adalah graf yang sisinya tidak mempunyai orientasi arah.
2. Graf berarah adalah setiap sisinya diberikan orientasi arah yang disebut dengan busur (arc). Dalam graf berarah, (u,v) dan (v,u)

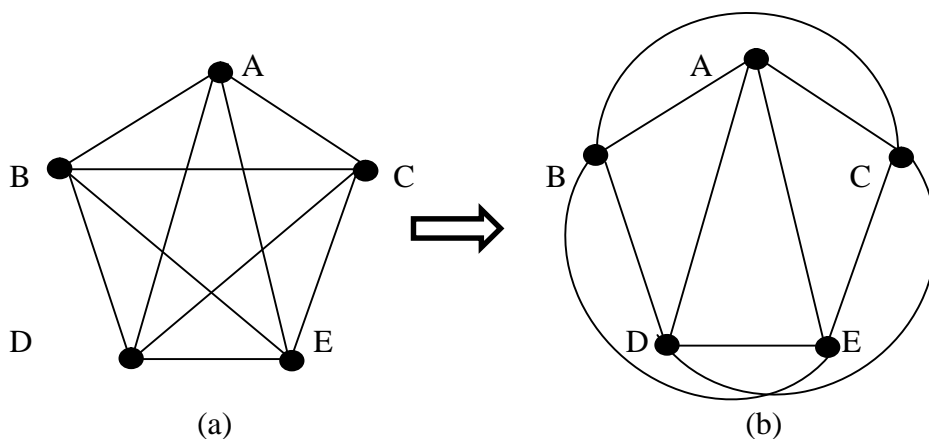
menyatakan dua buah busur yang berbeda. Pada busur (u,v) , simpul u dinamakan simpul asal dan simpul v merupakan simpul tujuan. Graf ini disebut graf berarah atau directed graf.

2.3 Graf Planar

Graf planar adalah suatu graf atau multigraf yang dapat digambarkan pada bidang (plane) sehingga sisi-sisinya tidak saling memotong (Lipschutz, 2008). Jika tidak, maka graf tersebut graf tidak planar (Munir, 2005). Sedangkan sebuah graf dikatakan graf bidang bila rusuk-rusuknya terletak pada bidang datar serta tidak saling berpotongan selain dititiknya (Wibisono, 2004). Berikut Gambar 2.4 dan Gambar 2.5 adalah contoh graf planar dan graf tidak planar :



Gambar 2.4 Graf Planar



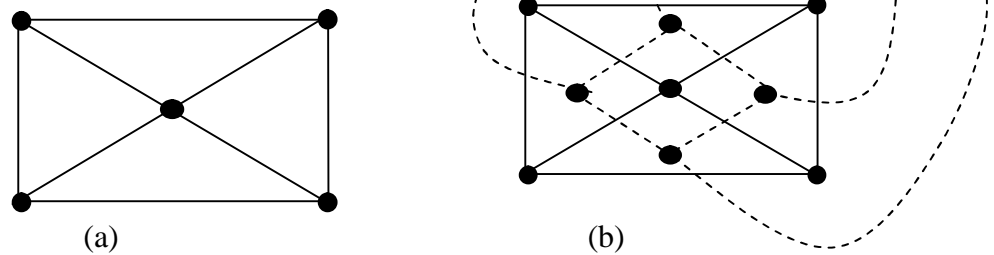
Gambar 2.5 Graf Tidak Planar

2.4 Graf Dual

Graf Dual adalah sebuah graf planar G yang direpresentasikan sebagai graf bidang, maka dapat dibuat suatu graf G^* yang secara geometri merupakan dual dari graf planar dengan cara berikut (Munir, 2007) :

1. Setiap wilayah atau muka (face) di G , buatlah sebuah simpul yang merupakan simpul untuk G^* .
2. Setiap sisi e di G , tariklah sisi e^* (yang menjadi sisi untuk G^*) yang memotong sisi e tersebut. Sisi e^* menghubungkan dua buah simpul v_1^* dan dipisahkan oleh sisi e di G .

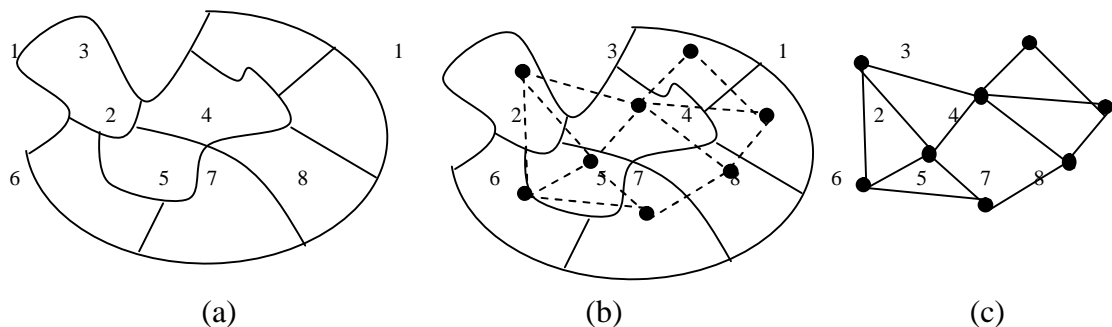
Berikut adalah contoh graf dual :



Gambar 2.6 Pembentukan Graf Dual

Sedangkan untuk merepresentasikan sebuah peta yang terdiri dari sejumlah wilayah memiliki sedikit perbedaan dengan graf dual yang telah disebutkan sebelumnya. Pada graf yang merepresentasikan peta bidang luar tidak dinyatakan sebagai sebuah simpul (Munir, 2005).

Berikut contoh graf yang merepresentasikan peta :



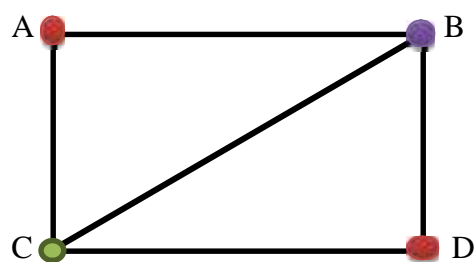
Gambar 2.7 Representasi Peta dalam Graf

2.5 Pewarnaan Graf

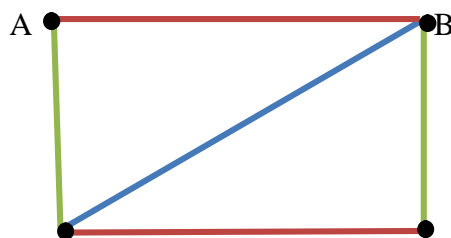
Pewarnaan suatu graf G adalah suatu pemberian warna pada salah satu elemen-elemennya (simpul dan sisi), sehingga elemen-elemen yang saling terhubung langsung mendapatkan warna yang berbeda. Ada tiga macam pewarnaan graf yaitu pewarnaan simpul, pewarnaan sisi, dan pewarnaan wilayah (region).

1. Pewarnaan simpul atau verteks adalah suatu pemberian warna untuk simpul dari G sedemikian sehingga simpul yang bersebelahan memiliki warna yang berbeda (Lipschutz, 2008)
2. Suatu pewarnaan sisi untuk graf G adalah suatu penggunaan sebagian atau semua warna untuk mewarnai semua sisi di G sehingga setiap pasang sisi yang mempunyai simpul persekutuan diberi warna yang berbeda.
3. Pewarnaan wilayah adalah warna yang diberikan ke setiap wilayah pada graf, sehingga tidak ada wilayah yang bersebelahan memiliki warna yang sama.

Berikut adalah contoh pewarnaan graf :

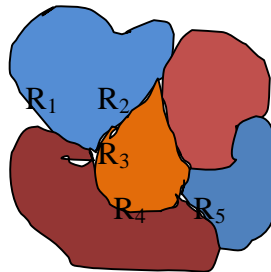


Gambar 2.8 Pewarnaan Simpul



C

D

Gambar 2.9 Pewarnaan Sisi**Gambar 2.10 Pewarnaan Wilayah**

2.6 Algoritma

2.6.1 Algoritma Welch Powell

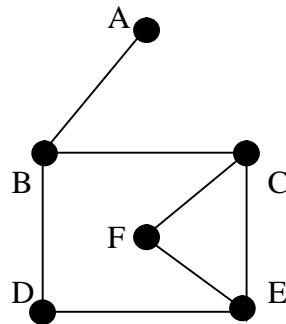
Algoritma Welch Powell dapat digunakan untuk mewarnai sebuah graf G secara efisien. Algoritma ini tidak selalu memberikan jumlah warna minimum yang diperlukan untuk mewarnai G , namun algoritma ini cukup praktis untuk digunakan dalam pewarnaan simpul sebuah graf. Algoritma Welch Powell hanya cocok digunakan untuk graf dengan orde yang kecil (As'ad, 2007).

Berikut adalah langkah-langkah algoritma Welch Powell (Lipschutz, 2008):

1. Urutkan simpul G menurut derajat yang mengecil.
2. Berikan warna pertama C_1 pada simpul pertama dan lalu secara berurutan, berikan C_1 ke setiap simpul yang tidak bersebelahan dengan simpul sebelumnya yang telah diberi C_1 .
3. Ulangi langkah 2 dengan warna ke dua C_2 dan simpul berikutnya yang belum diwarnai.
4. Ulangi langkah 3 dengan warna ketiga C_3 , lalu warna keempat C_4 dan sedemikian seterusnya sampai semua simpul telah diwarnai.

Berikut adalah contoh pewarnaan menggunakan algoritma Welch Powell :

Diberikan graf G pada Gambar 2.11 yang terdiri atas 6 buah simpul dan 7 buah sisi, akan ditentukan bilangan kromatiknya dengan menggunakan algoritma Welch Powell sebagai berikut:



Gambar 2.11 Pewarnaan Simpul yang Belum diwarnai Seluruh Simpulnya

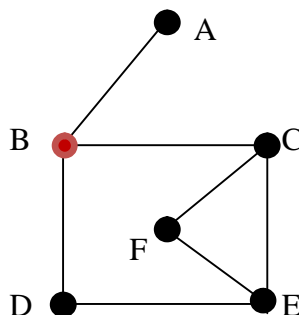
Penyelesaian pewarnaan simpul dengan algoritma Welch Powell sebagai berikut:

1. Mengurutkan simpul menurut derajatnya

Tabel 2.1 Derajat yang Telah Diurutkan dengan Algoritma Welch Powell

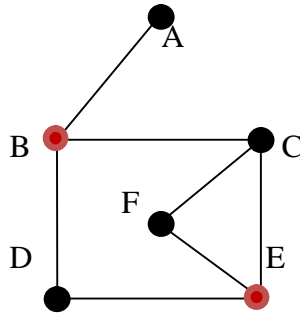
No	Simpul	Derajat
1	B	3
2	C	3
3	E	3
4	D	2
5	F	2
6	A	1

2. Berikan warna pertama C_1 pada simpul pertama dengan derajat tertinggi



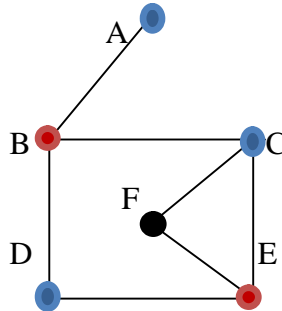
Gambar 2.12 Pewarnaan Simpul B Menggunakan Warna Merah

3. Berikan C_1 ke setiap simpul yang tidak bersebelahan dengan simpul sebelumnya yang telah diberi C_1



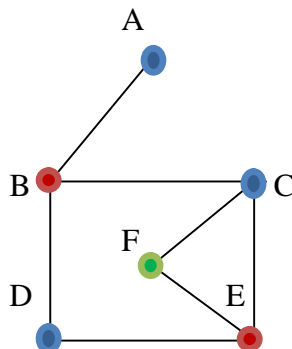
Gambar 2.13 Pewarnaan Simpul E Menggunakan Warna Merah

4. Ulangi langkah 2 dengan warna ke dua C_2 dan simpul berikutnya yang belum diwarnai.



Gambar 2.14 Pewarnaan Simpul A, C dan D Menggunakan Warna Biru

5. Ulangi langkah 3 dengan warna ketiga C_3 , lalu warna keempat C_4 dan sedemikian seterusnya sampai semua simpul telah diwarnai



Gambar 2.15 Pewarnaan Simpul F Menggunakan Warna Hijau

Dari Gambar 2.15 didapat 3 (tiga) warna minimum atau $\chi(G) = 3$

2.6.2 Algoritma Greedy

Algoritma greedy adalah algoritma yang memecahkan masalah langkah per langkah, pada setiap langkah membuat pilihan optimum (local optimum) pada setiap langkah dengan harapan bahwa langkah berikutnya mengarah ke solusi optimum global (global optimum). Algoritma greedy membuat keputusan berdasarkan pilihan yang ada sekarang, tidak melihat pilihan yang akan muncul kemudian. Karena itulah algoritma greedy dikategorikan dalam algoritma yang ‘berpandangan pendek’ dan tidak dapat diulang karena keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya (Adiwazsha).

Komponen algoritma Greedy terdiri dari :

a. Himpunan Kandidat C

Himpunan yang berisi elemen pembentuk solusi S .

b. Himpunan Solusi S

Himpunan yang berisi elemen solusi pemecahan masalah.

c. Fungsi Seleksi

Fungsi yang memilih kandidat yang paling memungkinkan dari himpunan kandidat untuk dimasukkan ke dalam himpunan solusi agar solusi optimal terbentuk. Kandidat yang sudah terpilih pada suatu langkah tidak akan dipertimbangkan lagi pada langkah selanjutnya. Fungsi seleksi terbagi 2, diantaranya :

1) Fungsi seleksi simpul

Memilih simpul yang memiliki sisi terbanyak atau derajat terbanyak.

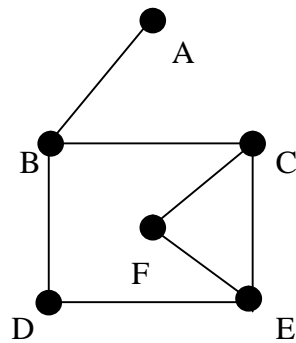
2) Fungsi seleksi warna

Memilih warna yang akan digunakan untuk mewarnai simpul. Pengerjaan fungsi seleksi warna terbagi atas dua langkah, yaitu :

- a. Jika layak, warna akan diambil dari himpunan solusi yaitu warna yang sudah digunakan sebelumnya.
 - b. Jika tidak ada satu pun warna dari himpunan solusi layak atau himpunan solusi masih kosong, maka akan diambil dari himpunan kandidat C yaitu warna yang belum pernah digunakan.
- d. Fungsi Kelayakan
- Fungsi yang memeriksa apakah suatu kandidat yang terpilih akan menimbulkan solusi yang layak, yaitu kandidat tersebut, bersama dengan himpunan solusi yang terpilih tidak akan melanggar kendala yang berlaku pada masalah.
- e. Fungsi Obyektif
- Fungsi yang memaksimalkan atau meminimalkan nilai solusi. Adapun langkah-langkah pewarnaan graf dengan menggunakan algoritma Greedy sebagai berikut:
1. Inisialisasi himpunan solusi dengan kosong.
 2. Melakukan pemilihan *vertex* yang akan diisi warnanya dengan fungsi seleksi *vertex*.
 3. Memilih kandidat warna dengan menggunakan himpunan kandidat kurangi warna anggota himpunan kandidat dengan warna yang diambil.
 4. Periksa kelayakan warna yang dipilih menggunakan langkah 3. Jika layak dimasukkan ke himpunan solusi.
 5. Periksa apakah solusi sudah meliputi seluruh *vertex*. Jika sudah maka berhenti, jika belum maka akan kembali ke langkah 3.

Berikut adalah contoh pewarnaan simpul menggunakan algoritma Greedy dengan graf yang sama :

Diberikan graf G pada Gambar 2.16 yang terdiri atas 6 buah simpul dan 7 buah sisi, akan ditentukan bilangan kromatiknya dengan menggunakan algoritma Greedy sebagai berikut:



Gambar 2.16 Pewarnaan Simpul yang Seluruh Simpulnya Belum Diwarnai

Penyelesaian pewarnaan simpul dengan algoritma Greedy sebagai berikut:

1. Mengurutkan simpul menurut derajatnya

Tabel 2.2 Derajat yang Telah Diurutkan dengan Algoritma Greedy

No	Simpul	Derajat
1	B	3
2	C	3
3	E	3
4	D	2
5	F	2
6	A	1

2. MatriksKetetanggan

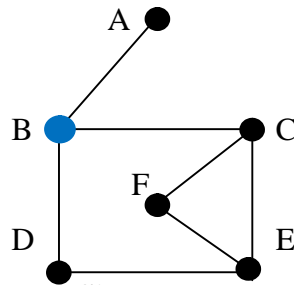
```

0  1  0  0  0  0
1  0  1  1  0  0
0  1  0  0  1  1
0  1  0  0  1  0
0  0  1  1  0  1
0  0  1  0  1  0

```

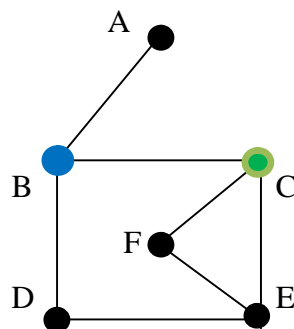
3. a. Himpunan kandidat C : {biru, hijau, merah, ungu, kuning, orange}
- b. Himpunan solusi S : { }
- c. Fungsi seleksi simpul : simpul B
- d. Fungsi seleksi Warna : Warna biru

- e. Fungsi kelayakkan :Dilihat dari matriks ketetanggannya simpul B bertetangga dengan 3 (tiga) buah simpul yang belum diwarnai. Maka birulayaldigunakan.
- f. Fungsi obyektif :Solusi belum optimal karena semua simpul belum diwarnai.



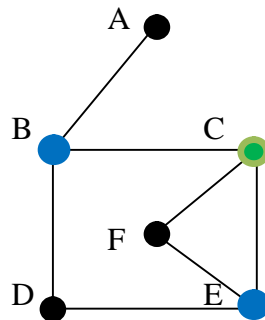
Gambar 2.17 Pewarnaan Simpul B Menggunakan Warna Biru

4. a. Himpunan kandidat C :{ hijau, merah, ungu, kuning, orange }
- b. Himpunan solusi S :{ biru }
- c. Fungsi seleksi simpul :simpul C
- d. Fungsi seleksi Warna :Warna biru
- e. Fungsi kelayakkan :Dilihat dari matriks ketetanggannya simpul C bertetangga dengan 3 (tiga) buah simpul dimana salah satu simpul telah diwarnai dengan warna biru. Maka warna biru tidak layak digunakan.
- f. Fungsi seleksi Warna :Warna hijau
- g. Fungsi kelayakkan :Dilihat dari matriks ketetanggannya, warna hijau layak digunakan karena tidak salah satu simpul tetangga menggunakan warna hijau. Maka warna hijau layak digunakan.
- h. Fungsi obyektif :Solusi belum optimal karena semua simpul belum diwarnai.



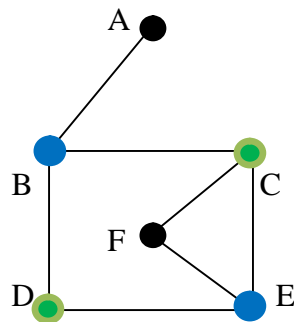
Gambar 2.18Pewarnaan Simpul C Menggunakan Warna Hijau

5. a. Himpunan kandidat C :{ merah, ungu, kuning, orange }
- b. Himpunan solusi S :{ biru, hijau }
- c. Fungsi seleksi simpul :simpul E
- d. Fungsi seleksi Warna :Warna biru
- e. Fungsi kelayakkan :Dilihat dari matriks ketetanggannya, warna biru layak digunakan karena tidak satu pun simpul tetangga simpul E menggunakan warna biru. Maka warna biru layak digunakan.
- f. Fungsi obyektif :Solusi belum optimal karena semua simpul belum diwarnai.



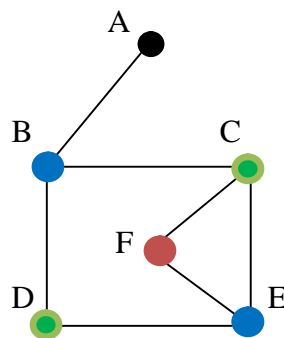
Gambar 2.19 Pewarnaan Simpul E Menggunakan Warna Biru

6. a. Himpunan kandidat C :{ merah, ungu, kuning, orange }
- b. Himpunan solusi S :{ biru, hijau }
- c. Fungsi seleksi simpul :simpul D
- d. Fungsi seleksi Warna :Warna biru
- e. Fungsi kelayakkan :Dilihat dari matriks ketetanggannya simpul D bertetangga dengan 2 (dua) buah simpul dimana salah satu simpul telah diwarnai dengan warna biru. Maka warna biru tidak layak digunakan.
- d. Fungsi seleksi Warna :Warna hijau
- e. Fungsi kelayakkan :Warna hijau layak digunakan karena simpul tetangga telah menggunakan warna biru.
- f. Fungsi obyektif :Solusi belum optimal karena semua simpul belum diwarnai



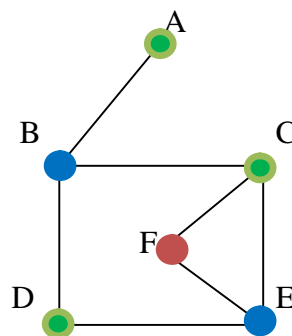
Gambar 2.20Pewarnaan Simpul D Menggunakan Warna Hijau

7. a. Himpunan kandidat C :{ merah, ungu, kuning, orange }
- b. Himpunan solusi S :{biru, hijau }
- c. Fungsi seleksi simpul :simpul F
- d. Fungsi seleksi Warna :Warna biru
- e. Fungsi kelayakkan :Dilihat dari matriks ketetanggannya simpul F bertetangga dengan 2 (dua) buah simpul dimana salah satu simpul telah menggunakan warna biru. Maka warna biru tidak layak digunakan.
- f. Fungsi seleksi Warna :Warna hijau
- g. Fungsi kelayakkan :Kedua simpul F telah menggunakan warna biru dan warna hijau. Maka hijau tidak layak digunakan.
- h. Fungsi seleksi Warna :Warna merah
- i. Fungsi kelayakkan :Semua simpul tetangga simpul F telah diwarnai dan tidak satupun simpul menggunakan warna merah. Maka merah layak digunakan.
- j. Fungsi obyektif :Solusi belum optimal karena semua simpul belum diwarnai



Gambar 2.21Pewarnaan Simpul F Menggunakan Warna Merah

8. a. Himpunan kandidat C : { ungu, kuning, orange }
- b. Himpunan solusi S : { biru, hijau, merah }
- c. Fungsi seleksi simpul : Simpul A
- d. Fungsi seleksi Warna : Warna biru
- e. Fungsi kelayakkan : Dilihat dari matriks ketetangannya simpul A bertetangga dengan sebuah simpul yang telah diwarnai dengan warna biru. Maka warna biru tidak layak digunakan.
- d. Fungsi seleksi Warna : Warna hijau
- e. Fungsi kelayakkan : Warna hijau layak digunakan karena simpul tetangga telah menggunakan warna biru.
- f. Fungsi obyektif : Solusi telah optimal karena semua simpul telah diwarnai



Gambar 2.22 Pewarnaan Simpul A Menggunakan Warna Hijau

Dari Gambar 2.22 didapat 3 (tiga) warna minimum atau $\chi(G) = 3$

2.7 Kompleksitas Algoritma

Sebuah program komputer, meskipun diturunkan dari sebuah algoritma yang benar, mungkin saja tidak berguna untuk jenis masukan tertentu karena waktu yang diperlukan untuk melakukan program atau memori yang diperlukan untuk menyimpan data, variabel program, dan lain sebagainya terlalu besar. Analisis algoritma mengacu pada proses memperoleh perkiraan dari waktu dan ruang yang diperlukan untuk mengeksekusi algoritma (Johnsonbaugh, 1997).

Pembahasan yang digunakan adalah kompleksitas algoritma karena kompleksitas ruang dan waktu membutuhkan ruang dan waktu yang dikaitkan dengan struktur data yang digunakan untuk mengimplementasikan algoritma.

Mengukur kompleksitas algoritma adalah menghitung banyaknya operasi yang dilakukan oleh algoritma.